EDUCAR PARA TRANSFORMAR

CAMADA DE TRANSPORTE

Prof: Flávio Alves

Tópicos

8.1 Definições Gerais

8.2 Os Modelos OSI e TCP/IP

8.3 Funções da Camada de Transporte

8.4 Portas de Comunicação

8.5 Sockets

8.6 Exemplos de portas bem conhecidas

8.7 Protocolos da Camada de Transporte

Definições Gerais

EDUCAR PARA TRANSFORMAR

A camada de Transporte (modelo OSI, camada 4) tem por **função fornecer um fluxo de dados entre dois hosts**, para a camada de aplicação. Permitir a comunicação entre duas máquinas.

Responde a requisições de serviço da camada de Sessão e emite requisições de serviço para camada de Rede.

Corresponde à mesma camada de Transporte no TCP/IP.

Os Modelos OSI e TCP/IP

Podemos descrever a pilha TCP/IP em termos do modelo OSI. Algumas funções da camada de sessão da OSI estão implementadas na camada de Transporte TCP/IP.

FUNÇÕES DA CAMADA DE TRANSPORTE

EDUCAR PARA TRANSFORMAR

1) Fornece **serviços de comunicação host a host** para aplicações;
2) Fornece serviços como **controle de fluxo**
3) Permite a **entrega de pacotes na mesma ordem em que foram enviados**
4) Permite **entrega confiável de pacotes** – se pacotes forem perdidos, podem ser reenviados
5) Multiplexação: **o uso de portas permite que várias aplicações no mesmo computador acessem a rede de forma simultânea**

PARADIGMA CLIENTE-SERVIDOR p1

EDUCAR PARA TRANSFORMAR

Embora existam varias **maneiras de se realizar a comunicação entre processos**, a mais comum delas é por meio do **paradigma cliente/servidor**. Um processo no **host local**, denominado **cliente**, **solicita serviços a outro processo**, normalmente localizado no host remoto, denominado **servidor**.

Ambos os processos (cliente e servidor) tern o mesmo nome. Por exemplo, para obtermos o dia e horário de uma maquina remota, **precisamos de um processo cliente denominado Daytime**, em execução no host local e **um processo servidor Daytime**, em execução em uma maquina remota.

PARADIGMA CLIENTE-SERVIDOR p2

EDUCAR PARA TRANSFORMAR

Os sistemas operacionais atuais oferecem suporte tanto a **ambientes multiusuários como multiprogramação**. Um computador remoto pode executar vários programas servidor simultaneamente, da mesma forma que computadores locais podem executar um ou mais programas cliente. Para a comunicação entre processos, devemos definir o seguinte:

1. **Host local**
2. **Processo local**
3. **Host remoto**
4. **Processo remoto**

ENDEREÇAMENTO p1

EDUCAR PARA TRANSFORMAR

Toda vez que precisarmos transmitir algo para um destino especifico entre varios existentes, necessitaremos de um endereço.

Na **camada de enlace de dados**, utilizamos o **endereço MAC** para **escolher um nó entre varios nos**, caso a conexao nao seja ponto a ponto. **Um frame** na camada de enlace precisa do endereço MAC de destino para sua entrega e de um endereço de origem para a resposta do destino.

Na **camada de rede**, utilizamos **o endereço IP para escolher um host** entre milhões. **Um datagrama na camada de rede** precisa de um endereço IP de destino para sua entrega e de um endereço IP de origem para a resposta do destino.

ENDEREÇAMENTO p2

EDUCAR PARA TRANSFORMAR

Na camada de transporte, utilizamos o endereço de camada de transporte, denominado **número de porta**, para escolher um entre os vários processos que estao em execução no host de destino.

O numero da porta de destino e necessário para entrega; o numero da porta de origem e necessário para resposta.

PORTAS DE COMUNICAÇÃO

EDUCAR PARA TRANSFORMAR

Usadas para determinar qual processo em um host local se comunica com qual processo em um host remoto.

Para um processo se comunicar com outro processo se utilizam portas.

A porta é um **número de 16 bits** (0 a 65.535), que identifica qual protocolo da camada superior devem entregar as mensagens recebidas.

As portas bem ser ser: bem conhecidas, registradas ou efêmeras.

As portas **são sempre associadas com um endereço IP e um protocolo de transporte**.

FAIXAS DE ENDEREÇOS

EDUCAR PARA TRANSFORMAR

A IANA (Internet Assigned Number Authority) dividiu o numero das portas em tres faixas: conhecidos, registrados e dinamicos (ou privados).

- **Portas conhecidas**. As portas na faixa de 0 a 1023 sao atribuidas e controladas pela IANA.

Estas sao as portas conhecidas (well-known port numbers).

- **Portas registradas ou efêmeras**. As portas na faixa de 1024 a 49151 não são atribuídas ou controladas pela IANA, são **processos ou aplicativos do usuário**. Elas podem ser registradas na IANA para impedir duplicação.

- **Portas dinâmicas**. As portas na faixa de 49152 a 65535 não são controladas nem registradas. Elas podem ser usadas por qualquer processo. Estas sao denominadas portas efêmeras.

Soquetes (Sockets)

EDUCAR PARA TRANSFORMAR

Um soquete **é um API (Application Programming Interface) para protocolos de comunicação.**

Usado por um processo **para solicitar serviços de rede do Sistema Operacional**

Composto por: **{protocolo, endereço, porta}**

Exemplo: {TCP, 192.168.22.53,80}

OU

192.168.22.53:80

Exemplos de Portas bem-conhecidas

EDUCAR PARA TRANSFORMAR

Algumas das portas mais comuns utilizadas:

Aplicação	Porta
FTP	21
SSH (comunicação segura criptografada)	22
Telnet (sem segurança)	23
DNS	53
DHCP	67,68
HTTP	80
IMAP (protocolo de correio)	143
HTTPS	443

Protocolos da camada de Transporte

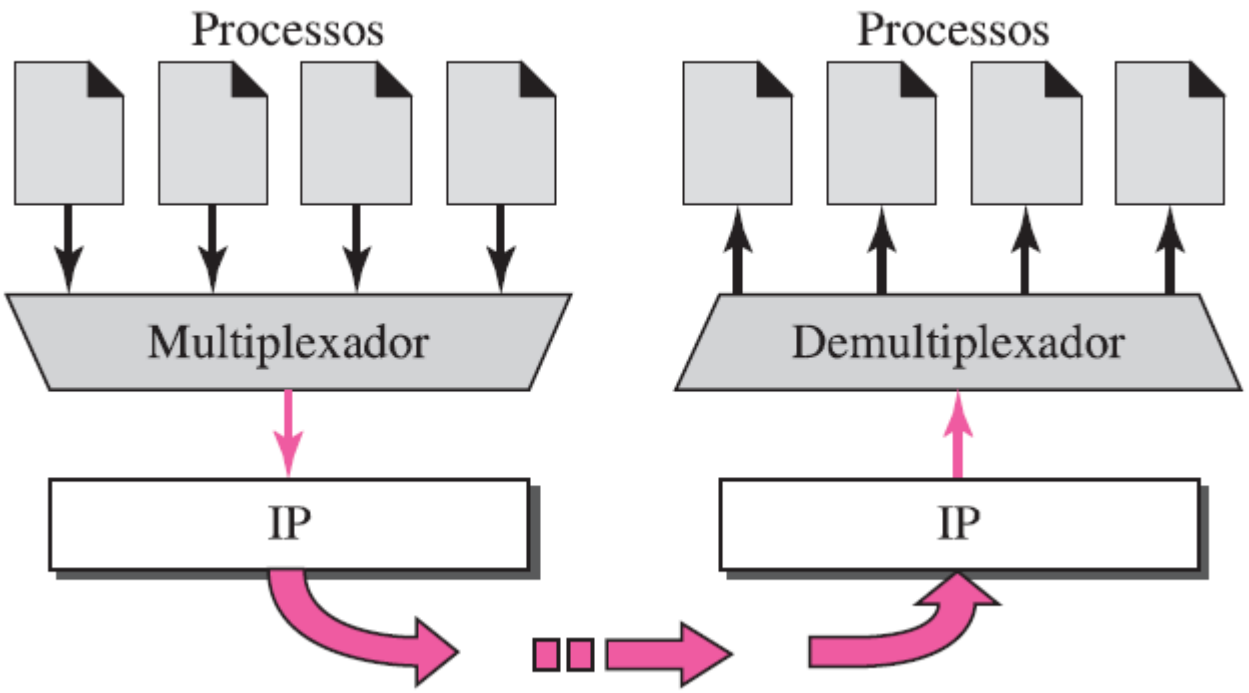
EDUCAR PARA TRANSFORMAR

Aplicação	Proto-colo
Fluxo de dados confiáveis entre dois hosts. A entrega de pacotes, chamada de segmentos, de um host a outro é garantida	TCP
Envia pacotes de dados chamados de datagramas, de um host a outro, sem garantida que os pacotes cheguem aos seus destinos.	UDP
Stream Control Transmission Protocol. **Orientado a mensagens como o UDP, com controle de congestionamento como o TCP**. Fornece **caminhos redundantes que aumentam a resiliência e a confiabilidade da conexão**. Nativo no FreeBSD v.7, portado para outros sistemas. (inclusive no Windows e no Mac OS)	SCTP

Multiplexação / Desmultiplexação

EDUCAR PARA TRANSFORMAR

Um mecanismo de endereçamento possibilita a camada de transporte multiplexar dados na origem provenientes de diferentes aplicações, transmiti-los por um circuito virtual e desmultiplexa-los no destino.



Multiplexação

EDUCAR PARA TRANSFORMAR

No lado do emissor, **podem existir vários processos que precisam transmitir pacotes**. Entretanto, ha somente um protocolo de camada de transporte em execução em dado instante.

Trata-se de uma relação varios-para-um e que requer a multiplexação.

O protocolo de transporte aceita mensagens provenientes de diferentes processos, diferenciados pelos números de porta a elas atribuídas.

Após acrescentar o cabeçalho, a camada de transporte passa o pacote para a camada de rede.

Desmultiplexação

EDUCAR PARA TRANSFORMAR

No lado do receptor, a relação e de um-para-varios e requer desmultiplexação.

A camada de transporte recebe os datagramas da camada de rede.

Após **a verificação de erros e a eliminação do cabeçalho**, a camada de transporte **entrega cada mensagem para o processo usuario apropriado baseado no numero da porta**.

U D P

UDP p1

EDUCAR PARA TRANSFORMAR

0 UDP (User Datagram Protocol) e um protocolo de **transporte sem conexao (connection less) e nao-confiavel.**

Ele nao adiciona nenhum controle adicional aos serviços de entrega do IP, exceto pelo fato de **implementar a comunicaçao entre processos, em vez da comunicação entre hosts**. Da mesma forma, a verificação de erros é implementada de forma muito limitada.

Se o UDP e tao simples assim, por que um processo iria querer usa-lo? Com as desvantagens vem algumas vantagens.

UDP p2

EDUCAR PARA TRANSFORMAR

O UDP é **um protocolo muito simples com um mínimo de overhead** (quantidade de bits que tem-se que incluir nos dados para que a informação chegue correta ao seu destino).

Se um processo quiser enviar uma pequena mensagem e não se preocupar muito com a confiabilidade, o UDP e uma boa escolha.

Enviar uma pequena mensagem através do UDP exige menor interação entre o emissor e o receptor do que quando usamos o TCP ou o SCTP.

PORTAS CONHECIDAS DO UDP

EDUCAR PARA TRANSFORMAR

Porta	Protocolo	Descrir;ao
7	Echo	Ecoa um datagrama recebido de volta para o emissor
9	Discard	Descarta qualquer datagrama recebido
11	Users	Usuarios ativos
13	Daytime	Retorna data e hora
17	Quote	Retorna um comentario do dia
19	Chargen	Retorna uma string de caracteres
53	Nameserver	Domain Name Services
67	BOOTPs	Servidor bootstrap
68	BOOTPc	Cliente bootstrap
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol

Datagramas de Usuário UDP

EDUCAR PARA TRANSFORMAR

Os **pacotes UDP, denominados datagramas de usuario**, possuem um cabeçalho de tamanho fixo, de 8 bytes.

Os campos de um datagrama são:

- ✓portas de origem,
- ✓portas de destino e
- ✓comprimento.

Porta de Origem

EDUCAR PARA TRANSFORMAR

Porta de origem. Esse campo especifica o número da porta usada pelo processo em execução no host de origem. Ele tem 16 bits de comprimento, significando que o numero da porta pode variar de O a 65535. **Se host de origem for um cliente** (um cliente enviando uma solicitação), **o numero da porta**, na maioria dos casos, e um **numero de porta efêmero** solicitado pelo processo cliente **é escolhido randomicamente pelo software UDP** em execução no host de origem. **Se host de origem for um servidor** (um servidor enviando uma resposta), **o número da porta**, na maioria dos casos, **é um número de porta conhecido**.

Porta de Destino

EDUCAR PARA TRANSFORMAR

Porta de destino. Esse campo especifica o número da porta usado pelo processo em execuçao no host de destino. Ele também tern 16 bits de comprimento. Se host de destino for um servidor (um cliente transmitindo uma solicitação), o número da porta, na maioria dos casos, e um número de porta conhecido. Se **o host de destino for um cliente** (um servidor transmitindo sua resposta), **o número da porta**, na maioria das vezes, **é um número de porta efemero**. Nesse caso, **o servidor copia o número de porta efemero que recebeu no pacote de solicitação.**

Comprimento

Comprimento. Esse campo de 16 bits define o comprimento total de um datagrama UDP, compreendendo cabeçalho mais dados.

Os 16 bits podem definir um comprimento total entre O a 65.535 bytes.

Entretanto, o comprimento total deve ser menor, pois um datagrama UDP deve ser repassado em um datagrama IP de comprimento total igual a 65.535 bytes.

O campo de comprimento em um datagrama UDP, na verdade, e desnecessário.

CheckSum

EDUCAR PARA TRANSFORMAR

Checksum. Esse campo de 16 bits e usado para detectar erros na transmissao de datagrama UDP (cabeçalho mais dados).

T C P

TCP

O segundo protocolo da camada de transporte é denominado TCP (Transmission Control Protocol) que, assim como o UDP, e um protocolo de comunicação entre processos finais (programa a programa). Portanto, o TCP, bem como o UDP, usa numeros de portas. **Diferentemente do UDP, o TCP e um protocolo orientado a conexao; ele cria uma conexao virtual entre dois processos TCPs para a transmissao de dados.** Alem disso, o TCP implementa mecanismos de controle de fluxo e de erros na camada de transporte.

T C P – comunicação entre processos

EDUCAR PARA TRANSFORMAR

Assim como o UDP, o TCP implementa a comunicação entre processos utilizando números de porta.

Em suma, o TCP e um protocolo de transporte orientado a conexao e confiavel.

Ele acrescenta funcionalidades aos serviços IP orientados a conexao e confiabilidade .

T C P – comunicação entre processos

EDUCAR PARA TRANSFORMAR

Porta	Protocolo	Descrição
7	Echo	Ecoa um datagrama recebido de volta para o emissor
9	Discard	Descarta qualquer datagrama recebido
11	Users	Usuarios ativos
13	Daytime	Retoma a data e a hora
17	Quote	Retorna o comentario do dia
19	Chargen	Retoma uma string de caracteres
20	FTP, Dados	File Transfer Protocol (conexiio de dados)
21	FTP, Control	File Transfer Protocol (conexiio de controle)
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Narne Services
67	BOOTP	Bootstrap protocol
79	Finger	Finger
80	HTTP	HyperText transfer protocol
111	RPC	Remote Procedure Call

Serviço de entrega de fluxo de dados

EDUCAR PARA TRANSFORMAR

0 TCP, diferentemente do UDP, é um protocolo orientado a fluxo de dados. No UDP, um processo (um programa aplicativo) envia mensagens, com delimitadores predefinidos, para o UDP, para serem transmitidas. O UDP acrescenta seu próprio cabeçalho a cada uma dessas mensagens e as entrega para transmissão pelo IP.

Cada mensagem do processo e denominada datagrama de usuario e se torna, finalmente, um datagrama IP. Nern o IP nem o UDP estabelecem relação entre os datagramas.

O TCP, por outro lado, possibilita a um processo aplicativo enviar dados na forma de um fluxo de bytes e possibilita ao processo de recepção receber dados na forma de um fluxo de bytes.

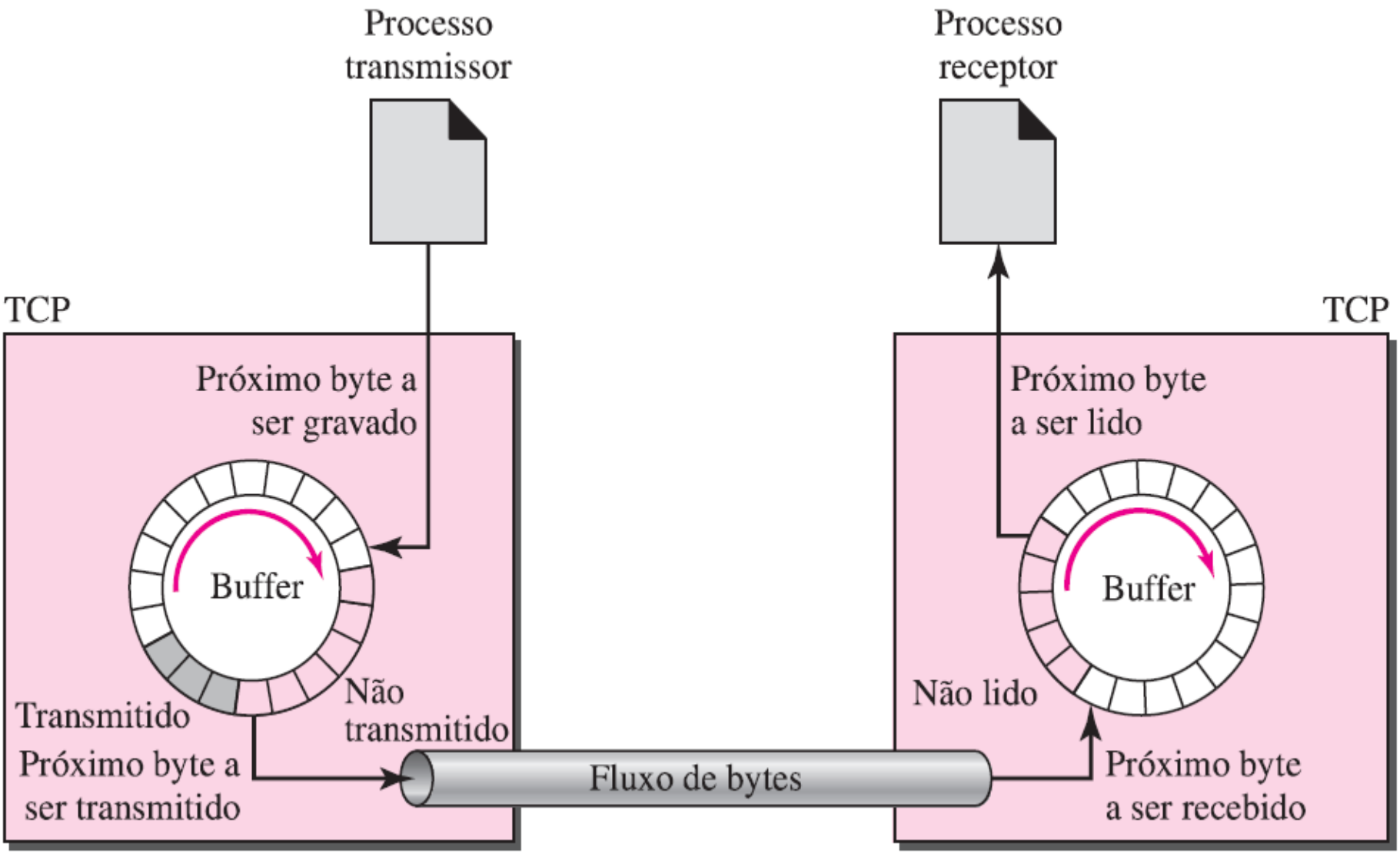
0 TCP cria um ambiente no qual os dois processos parecem estar conectados por um "canal" imaginario que transporta seus dados pela Internet.

Buffers de Transmissão e Recepção

EDUCAR PARA TRANSFORMAR

Como existe a possibilidade dos processos de trans missao e de recepçao nao gerarem ou lerem dados em uma mesma velocidade, o TCP precisa de buffers para seu armazenamento. **Existem dois buffers, um buffer de transmissao e um buffer de recepçao, um em cada direçao**. (**Esses buffers tambem sao necessarios para os mecanismos de controle de fluxo e de erros usados pelo TCP**.) Uma forma de implementar um buffer é usar uma matriz circular de posições (1 byte) conforme mostrado na Figura. Para fins de simplificação , mostramos dois buffers de 20 bytes cada; normalmente, os buffers sao implementados com capacidade de centenas ou milhares de bytes, de pendendo da implementa9ao. Usamos tambem buffers de mesmo tamanho, o que nem sempre e verdade na pratica.

Buffers de Transmissão e Recepção



Buffers de Transmissão e Recepção

EDUCAR PARA TRANSFORMAR

A Figura ilustra a movimentaçao de dados em uma direçao. No lado do emissor, **o buffer possui tres tipos de camaras**. A parte em **branco contem camaras vazias** que podem ser preenchidas pelo processo transmissor (produtor). **A area cinza armazena bytes que foram enviados**, **mas ainda nao confirmados pelo receptor**. O TCP mantem esses bytes no buffer ate receber uma confirmaçao. **A area colorida contem bytes a serem enviados pelo transmissor TCP**. Entretanto, o TCP é capaz de enviar apenas parte dos dados desta seçao colorida. Isso poderia se dar em decorrencia da lentidao no processo de recepçao ou quem sabe em virtude de congestionamentos na rede.

Perceba tambem que após os bytes nas camaras cinza serem confirmados, as camaras sao limpas e ficam disponíveis para uso pelo processo de transmissao. E por essa razao que implementamos um buffer circular

Buffers de Transmissão e Recepção

EDUCAR PARA TRANSFORMAR

A operaçao do buffer no lado do receptor e mais simples. **O buffer circular e dividido em duas areas (segmentos brancos e coloridos). O segmento branco possui camaras vazias a serem preenchidas por bytes recebidos da rede**. As **seções coloridas contem bytes recebidos que podem ser lidos pelo processo receptor**. Quando um byte e lido pelo processo receptor, a camara e limpa e acrescentada ao pool de camaras vazias.

Segmentos

EDUCAR PARA TRANSFORMAR

Embora o sistema de buffers trate da disparidade entre as velocidades dos processos de geração e leitura de dados, precisamos de mais um passo antes de podermos efetivamente transmitir os dados. **A camada IP, como provedora de serviços para o TCP, precisa enviar dados em pacotes, não na forma de um fluxo de bytes.** Na camada de transporte, o TCP agrupa determinado número de bytes em pacotes, denominados segmento.

Segmentos p2

EDUCAR PARA TRANSFORMAR

O TCP acrescenta um cabeçalho a cada segmento (para fins de controle) e entrega o segmento para a camada IP para sua transmissão. Os segmentos são encapsulados em datagramas IP e transmitidos. Toda essa operação e transparente para o processo de recepção. Posteriormente, veremos que um segmento pode ser recebido fora de ordem, ser perdido ou corrompido e, então, precisa ser reenviado. Todas essas tarefas são implementadas pelo TCP sem que o processo receptor tome conhecimento dessas atividades

Serviço orientado à conexão p1

EDUCAR PARA TRANSFORMAR

0 TCP, diferentemente do UDP, e um protocolo orientado a conexão. Quando um processo no ponto A quer enviar e receber dados de outro processo no ponto B, ocorre o seguinte:

1. Os dois processos TCPs estabelecem uma conexao entre eles.
2. Os dados sao trocados em ambos os sentidos.
3. A conexao e encerrada.

Serviço orientado à conexão p2

EDUCAR PARA TRANSFORMAR

Esta e uma conexao virtual, e nao uma conexão física. **Um segmento TCP e encapsulado em um datagrama IP e pode ser recebido fora de ordem, perdido ou corrompido e, em seguida, precisa ser reenviado.** Cada um deles pode ser transmitido por um caminho diferente ate atingir o destino. Nao existe uma conexao física.

O TCP cria um ambiente orientado a fluxo de dados no qual ele assume a responsabilidade de entregar na ordem correta os bytes para o outro ponto.

A situação e semelhante a criar uma ponte que abarca varias ilhas, passando todos os bytes de uma ilha para outra em um unico sentido.

Serviço confiável

EDUCAR PARA TRANSFORMAR

0 TCP e um protocolo de transporte confiável.

Ele implementa mecanismos de confirmação para validar a chegada segura dos dados. Para fornecer um serviço de transporte confiável, o TCP implementa controle de erros. Embora o controle de erros considere o segmento como a unidade de dados para fins de detecção de erros (segmento corrompido ou perdido), **o controle de erros e orientado a bytes.**

Controle de Congestionamento p1

EDUCAR PARA TRANSFORMAR

Pode ocorrer congestionamento em uma rede, caso a carga na rede - o numero de pacotes enviados para a rede - seja maior que a capacidade da rede - o numero de pacotes que a rede e capaz de tratar.

Controle de congestionamento refere-se as tecnicas e mecanismos que podem tanto impedir o congestionamento, antes de ele acontecer, quanto eliminar o congestionamento, após este ter ocorrido. Em geral, podemos dividir os mecanismos de controle de congestionamento em duas grandes categorias: **controle de congestionamento de anel aberto (prevenção) e controle de congestionamento de anel fechado (eliminação).**

Controle de Congestionamento p2

EDUCAR PARA TRANSFORMAR

Ocorre congestionamento em uma rede ou internetwork porque os roteadores e comutado res possuem filas - buffers que retem os pacotes antes e depois do processamento. Suponha, por exemplo, que um roteador tenha uma fila de entrada e uma fila de saída para cada interface. Quando chega um pacote na interface de entrada, ele passa por tres etapas antes de partir.

Filas de um roteador

EDUCAR PARA TRANSFORMAR

1. 0 pacote e colocado no final da fila de entrada enquanto espera ser verificado.
2. 0 modulo de processamento do roteador elimina o pacote da fila de entrada assim que ele atingir a frente da fila e usa sua tabela de roteamento e o endereço de destino para determinar a rota.
3. 0 pacote e colocado na fila de saída apropriada e aguarda sua vez de ser transmitido

2 questões importantes

EDUCAR PARA TRANSFORMAR

Primeiro, **se a velocidade de chegada dos pacotes for maior que a velocidade de processamento dos pacotes**, **as filas de entrada se tomam cada vez mais longas**. Em segundo lugar, **se a velocidade de partida dos pacotes for menor que a velocidade de processamento dos pacotes**, as **filas de saída** se tornam cada vez mais longas.

Controle de Congestionamento Anel Aberto

No controle de congestionamento de anel aberto, as políticas são aplicadas para evitar o congestionamento antes que ele aconteça.

Entre as políticas de controle de congestionamento de anel aberto têm-se a política de retransmissão, a política de janelas, a política de confirmação, a política de descarte e a política de admissão.

Controle de Congestionamento Anel Fechado

EDUCAR PARA TRANSFORMAR

Mecanismos de controle de congestionamento de anel fechado tentam amenizar o congestionamento após este ter ocorrido.

Entre as políticas de controle de congestionamento de anel fechado têm-se a contrapressão, o pacote de controle, a sinalização implícita e a sinalização explícita.

S C T P

SCTP

EDUCAR PARA TRANSFORMAR

Combina as melhores características do UDP e do TCP. O SCTP e um protocolo **orientado a mensagens e confiável**. Ele **preserva os delimitadores das mensagens** e, ao mesmo tempo, detecta mensagens perdidas, duplicadas ou fora de ordem. Ele **também implementa mecanismos de controle de congestionamento e de fluxo**. O SCTP possui outros recursos inovadores não existentes no UDP e no TCP

Múltiplos fluxos de dados

EDUCAR PARA TRANSFORMAR

O TCP e um protocolo orientado a fluxo de dados. Cada conexao entre um cliente TCP e um servidor TCP envolve um unico fluxo de dados. O problema dessa abordagem e que a perda em qualquer ponto do fluxo de dados bloqueia a entrega do restante dos dados.

Isso pode ser aceitavel quando estamos transferindo textos, mas nao quando transmi timos dados em tempo real, por exemplo, em audio ou vídeo.

0 SCTP implementa serviços de transporte com múltiplos fluxos de dados em cada conexão, que e denominado associação na terminologia do SCTP.

Multihoming

EDUCAR PARA TRANSFORMAR

Uma conexão TCP envolve um unico endereço IP de destino e de origem. Isso significa que mesmo se o emissor ou o receptor for um host multihoming (conecta mais de um endereço físico a vários endereços IP), apenas um desses endereços IP por ponto final pode ser utilizado para estabelecer uma conexão.

È usar múltiplos pontos de conexão para evitar a queda da rede.

Por outro lado, uma associação no SCTP suporta serviços multihoming. Os hosts emissor e receptor podem definir vários endereços IP em cada ponto final, parte de uma associação.

Multihoming p2

EDUCAR PARA TRANSFORMAR

Nesse método, tolerante a falhas, quando uma rota falha, podemos usar outra interface para a entrega de dados, sem sofrer interrupção.

Esse método, tolerante a falhas, e extremamente útil quando estamos transmitindo e recebendo dados em tempo real, como no caso de telefonia via Internet.